

**WHAT IS CLAIMED IS:**

- 8/10/07
1. A computer-implemented data processing method, comprising:  
running a first process in a first address space and a second process in a second address space, the first process including a request to send to the second process data having a data type;  
*CREATE AN OBJECT.*  
calling at runtime a type creation function to create a first type object describing the data type, the first type object having a set of associated functions for processing data having the data type, the set of associated functions including a marshalling function for encoding data having the data type and an unmarshalling function for decoding data having the data type; and  
sending the data to the second process by executing the marshalling function on the data in the first process to generate encoded data and executing the unmarshalling function on the encoded data to decode the encoded data in the second process.
2. The method of claim 1, wherein the set of associated functions for processing data having the data type includes a print function for printing data having the data type.
3. The method of claim 1, wherein the data type is an array type, an integer type, a pointer type, a real type, a string type or a structure type.
4. The method of claim 1, wherein the first type object is a parameterized type object including an element identifying a location in memory, the parameterized type object describing a format for the data type based on one or more type parameters in the identified location.
5. The method of claim 4, wherein the element identifying a location in memory is an offset element identifying a location in memory relative to the data.
6. The method of claim 5, wherein the parameterized type object describes a dynamically sized array and the type parameters include data specifying a size of the dynamically sized array.

7. The method of claim 5, wherein the parameterized type object describes a dynamically typed pointer and the type parameters include data identifying a second type object.
8. The method of claim 1, wherein:  
the set of associated functions includes a type description function operable to generate a type object description describing the first type object;  
the encoded data includes an encoded representation of the type object description;  
and  
executing the unmarshalling function to decode the encoded data includes reconstructing the data in the second address space based on the type object description.
9. The method of claim 1, wherein:  
the first type object has a set of properties including a limitation condition specifying a limitation on permissible values for data having the data type; and  
executing the unmarshalling function to decode the encoded data includes returning an error message if the data violates the limitation condition.
10. The method of claim 1, wherein:  
the type creation function is called in the first process to create a first instance of the first type object and in the second process to create a second instance of the first type object.
11. The method of claim 1, wherein:  
the data has a first format in the first process; and  
the encoded data is decoded in the second process to generate data having a second format, the second format being different than the first format.
12. The method of claim 11, wherein:  
the encoded data is generated in a format that is independent of the first and second formats.

13. A computer-implemented data processing method, comprising:

running a first process in a first format in a first address space and a second process in a second format in a second address space, the first process including a request to send to the second process data having a data type;

calling at runtime in the first process a type creation function to create a first instance of a type object describing the data type, the type object having a set of associated functions for processing data having the data type, the set of associated functions including a marshalling function for encoding data having the data type and an unmarshalling function for decoding data having the data type;

calling at runtime in the second process the type creation function to create a second instance of the type object describing the data type; and

sending the data to the second process by executing in the first process the marshalling function of the first instance of the type object to generate encoded data in a format independent of the first and second formats, communicating the encoded data from the first process to the second process, and executing in the second process the unmarshalling function of the second instance of the type object to decode the encoded data.

14. A computer program product, tangibly stored on a computer-readable medium, for processing data, comprising instructions operable to cause a programmable processor to:

run a first process in a first address space and a second process in a second address space, the first process including a request to send to the second process data having a data type;

call at runtime a type creation function to create a first type object describing the data type, the first type object having a set of associated functions for processing data having the data type, the set of associated functions including a marshalling function for encoding data having the data type and an unmarshalling function for decoding data having the data type; and

send the data to the second process by executing the marshalling function on the data in the first process to generate encoded data and executing the unmarshalling function on the encoded data to decode the encoded data in the second process.

15. The computer program product of claim 14, wherein the set of associated functions for processing data having the data type includes a print function for printing data having the data type.

16. The computer program product of claim 14, wherein the data type is an array type, an integer type, a pointer type, a real type, a string type or a structure type.

17. The computer program product of claim 14, wherein the first type object is a parameterized type object including an element identifying a location in memory, the parameterized type object describing a format for the data type based on one or more type parameters in the identified location.

18. The computer program product of claim 17, wherein the element identifying a location in memory is an offset element identifying a location in memory relative to the data.

19. The computer program product of claim 18, wherein the parameterized type object describes a dynamically sized array and the type parameters include data specifying a size of the dynamically sized array.

20. The computer program product of claim 18, wherein the parameterized type object describes a dynamically typed pointer and the type parameters include data identifying a second type object.

21. The computer program product of claim 14, wherein:  
the set of associated functions includes a type description function operable to generate a type object description describing the first type object;  
the encoded data includes an encoded representation of the type object description;  
and

the unmarshalling function includes instructions operable to cause the programmable processor to reconstruct the data in the second address space based on the type object description.

22. The computer program product of claim 14, wherein:

the first type object has a set of properties including a limitation condition specifying a limitation on permissible values for data having the data type; and

the unmarshalling function includes instructions operable to cause the programmable processor to return an error message if the data violates the limitation condition.

23. The computer program product of claim 14, wherein:

the type creation function is called in the first process to create a first instance of the first type object and in the second process to create a second instance of the first type object.

24. The computer program product of claim 14, wherein:

the data has a first format in the first process; and  
the encoded data is decoded in the second process to generate data having a second format, the second format being different than the first format.

25. The computer program product of claim 24, wherein:

the encoded data is generated in a format that is independent of the first and second formats.

26. A computer program product, tangibly stored on a computer-readable medium, for processing data, comprising instructions operable to cause a programmable processor to:

run a first process in a first format in a first address space and a second process in a second format in a second address space, the first process including a request to send to the second process data having a data type;

call at runtime in the first process a type creation function to create a first instance of a type object describing the data type, the type object having a set of associated functions for processing data having the data type, the set of associated functions including a marshalling

function for encoding data having the data type and an unmarshalling function for decoding data having the data type;

call at runtime in the second process the type creation function to create a second instance of the type object describing the data type; and

send the data to the second process by executing in the first process the marshalling function of the first instance of the type object to generate encoded data in a format independent of the first and second formats, communicating the encoded data from the first process to the second process, and executing in the second process the unmarshalling function of the second instance of the type object to decode the encoded data.

add  
a<sup>3</sup> 7